Autolab Software Development Kit 1.10

User manual

Table of contents

1 – Introduction 5
2 – Install Autolab SDK 7
2.1 – USB Drivers installation 8
2.1.1 – Windows 8, 7 and Windows Vista 8
2.1.2 – Windows XP 12
3 – Autolab instrument 17
3.1 – Connect/Disconnect Autolab instrument 17
4 – Procedure 19
4.1 – Procedure model 19
4.2 – Using procedures 21
5 – LabVIEW 25
5.1 – Install LabVIEW examples 25
5.2 – Using Autolab SDK in LabVIEW 25
6 – Appendix 29
6.1 – Command names 29
6.2 – Input command parameter names 29
6.3 – Output command parameter names 30
6.4 – Command parameter types 30
6.5 – General remarks using individual modules 31
Acknowledgements 33

Autolab SDK

1 – Introduction

The Autolab Software Development Kit (Autolab SDK) is designed to control the Autolab instrument from different external applications such as LabVIEW, Visual Basic for Applications (VBA), scripting etc. With the Autolab SDK the external application can be used to measure complete procedures or control individual Autolab modules.

In order to use the Autolab SDK from other applications, these applications must have the possibility to use **.NET** assemblies or in the case of "older" applications to use **COM** assemblies. How to integrate these assemblies is explained in the manual of the application.

The Autolab SDK is compatible with NOVA however it does not require Nova to be installed. It is assumed that the user is familiar with Nova. More information about NOVA can be found in the NOVA User manual.

Note

Before reading this manual, we recommend you read Section 6.5 – *General remarks using individual modules* located at the end of this document.

Warning

The Autolab instrument is equipped with a dedicated embedded processor which is designed to control high precision and accurate timed measurement. The measuring techniques are using this embedded processor. Programming the individual Autolab modules from a windows environment is possible however the timing is poor.

2 – Install Autolab SDK

Follow these steps to install the Autolab SDK:

- Insert the Autolab SDK CD.
- Double click on Setup.exe.
- Follow the steps in the setup wizard.

When asked to install the *Shared Add-in Support Update for Microsoft .NET Framework (KB908002)* component, click Install.

When asked to install the *Microsoft .NET Framework* component, click Install. The Setup installs the following items:

On Windows XP:

- Hardware Setup Files and adk.x file:
  `C:\Program Files\Metrohm Autolab\Autolab SDK 1.10\Hardware Setup Files`
- Standard Procedures:
  `C:\Program Files\Metrohm Autolab\Autolab SDK 1.10\Standard Procedures`
- EcoChemie.Autolab.sdk.dll and other dll's:
  `C:\Program Files\Metrohm Autolab\Autolab SDK 1.10`
- Hardware Setup used by the Autolab SDK:
  `C:\Documents and Settings\All users\Application Data\Metrohm Autolab`
- USB driver:
  `C:\Program Files\Common Files\Metrohm Autolab\Drivers\Usb`
- LabVIEW Autolab examples:
  `C:\Program Files\Metrohm Autolab\Autolab SDK 1.10`

On Vista, Windows 7 and Windows 8:

- Hardware Setup Files and adk.x file
  `C:\Program Files\Metrohm Autolab\Autolab SDK 1.10\Hardware Setup Files`
- Standard Procedures:
  `C:\Program Files\Eco Chemie\Autolab SDK 1.10\Standard Procedures`
- EcoChemie.Autolab.sdk.dll and other dll's:
  `C:\Program Files\Metrohm Autolab\Autolab SDK 1.10`
- Hardware Setup used by the Autolab SDK:
  `C:\ProgramData\Metrohm Autolab`
- USB driver:
  `C:\Program Files\Common Files\Metrohm Autolab\Drivers\Usb`
- LabVIEW Autolab examples:
  `C:\Program Files\Eco Chemie\Autolab SDK 1.10`

To use the Autolab SDK in Office 2003, copy all the files with .dll extension from `C:\Program Files\Metrohm Autolab\Autolab SDK 1.10` folder to the `C:\Program Files\Microsoft Office\OFFICE11` folder.

2.1 – USB Drivers installation[1]

Note

Installation of the USB drivers is only required if no previous installation of Autolab Software is present on the computer.

2.1.1 – Windows 8, 7 and Windows Vista

After Autolab SDK has been successfully installed, connect the Autolab instrument to the computer using an available USB port. Switch on the instrument. If no Autolab USB drivers are installed, Windows will attempt to find a suitable driver for the instrument. Since the Autolab is not automatically recognized by Windows, the driver installation will fail, indicating that no driver was found this instrument (see Figure 2.1).

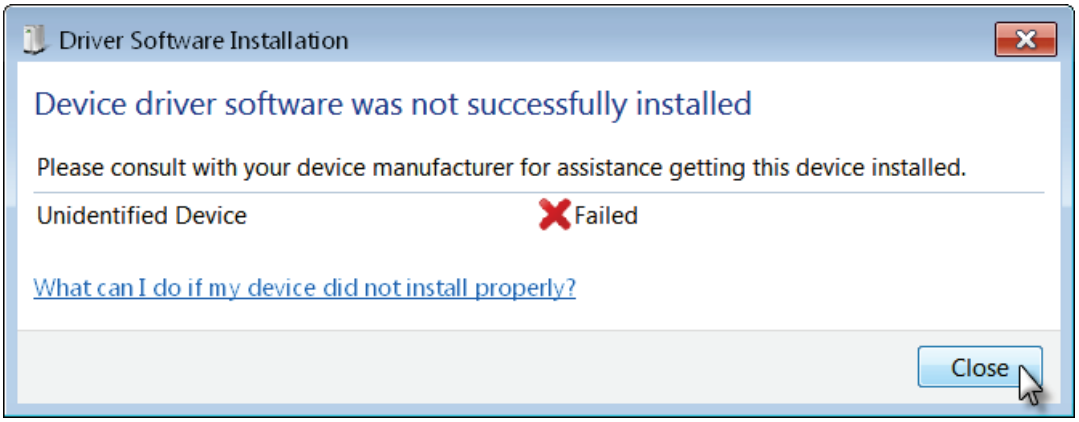

Driver Software Installation
Device driver software was not successfully installed
Please consult with your device manufacturer for assistance getting this device installed.
Unidentified Device
Failed
What can I do if my device did not install properly?
Close

Figure 2.1 – The driver installation dialog in Window 7 and Windows Vista

Click the Close button to close this message.

Open the Device Manager window, available in the Control Panel. The Autolab will be listed as **Unknown device** under Other devices (see Figure 2.2).

[1] Depending on the instrument, this part of the installation process may have to be repeated twice.

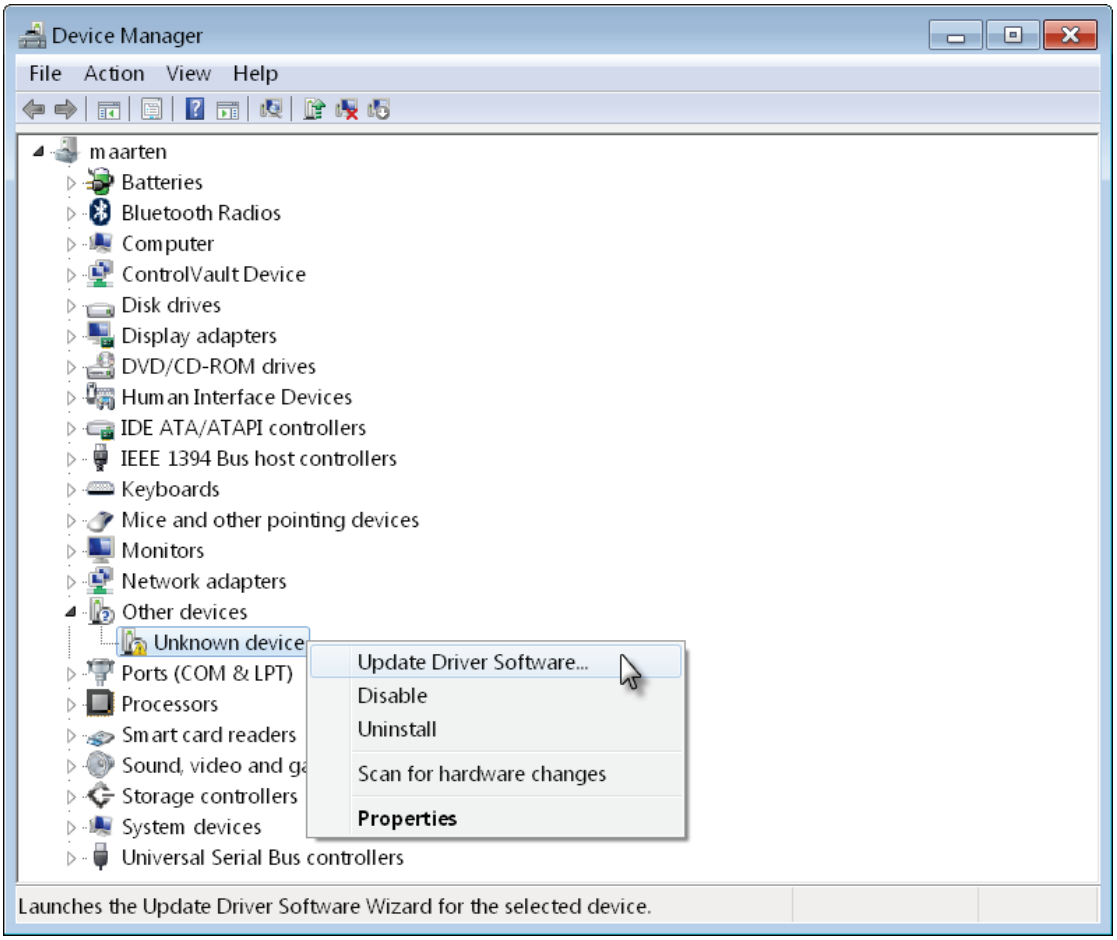


Figure 2.2 – The Autolab is listed as Unknown device in the Device Manager

Right-click the Unknown device in the Device Manager and select the **Update Driver Software** option from the context menu (see Figure 2.2).

The Update Driver Software dialog will be displayed (see Figure 2.3).

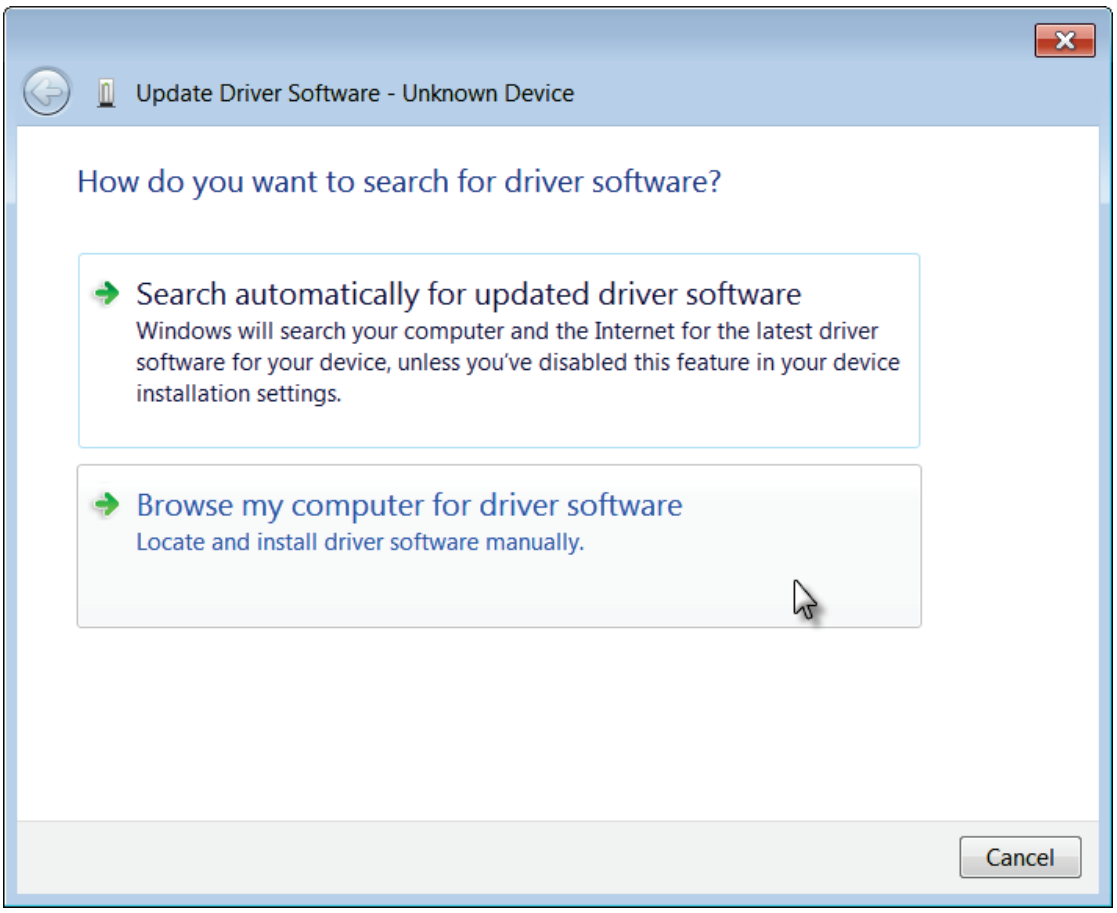

Update Driver Software - Unknown Device
How do you want to search for driver software?
Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.
Browse my computer for driver software
Locate and install driver software manually.
Cancel

Figure 2.3 – Installing the driver for the Autolab (1/4)

Select the second option, *Browse my computer for driver software* to continue. In the next screen, click the Browse... button to specify the location of the Autolab drivers. In the Browse for Folder window, navigate to the `C:\Program Files\Common Files\Metrohm Autolab\Drivers\Usb` folder and click OK to continue (see Figure 2.4)

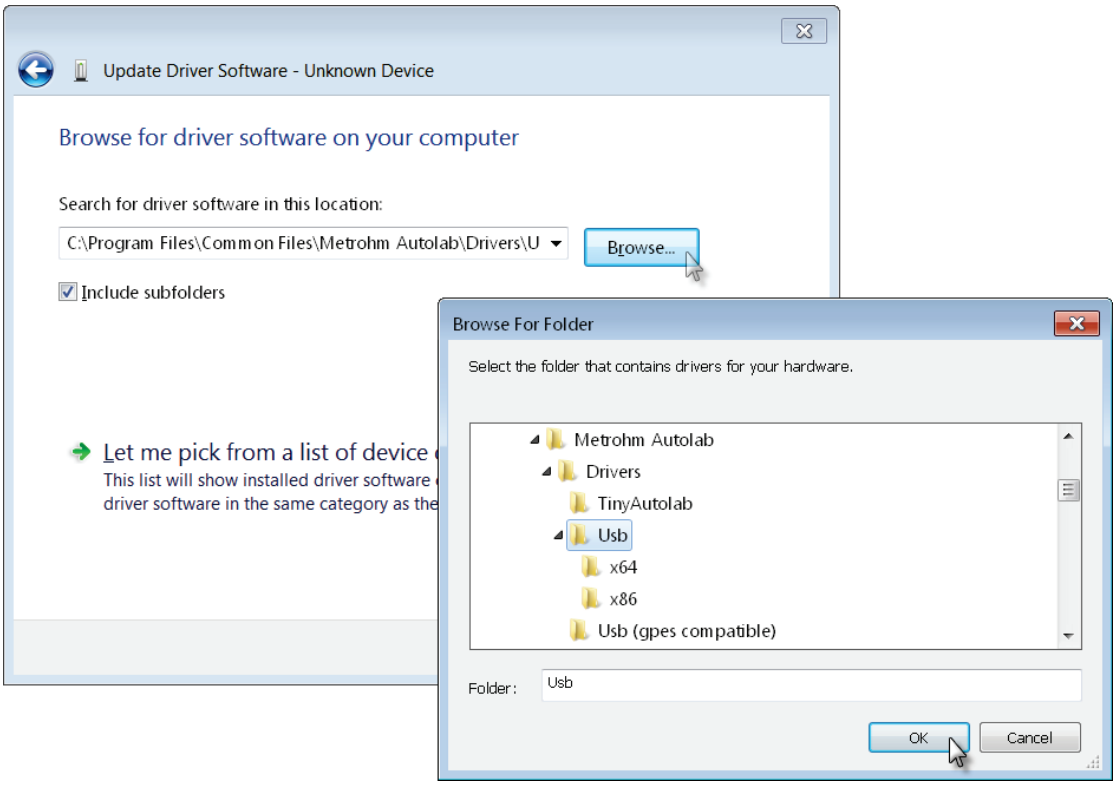


Figure 2.4 – Installing the driver for the Autolab (2/4)

Click OK to confirm the location of the driver and the Next to proceed with the installation. The Autolab driver is not recognized by Windows and a warning will be displayed (see Figure 2.5).

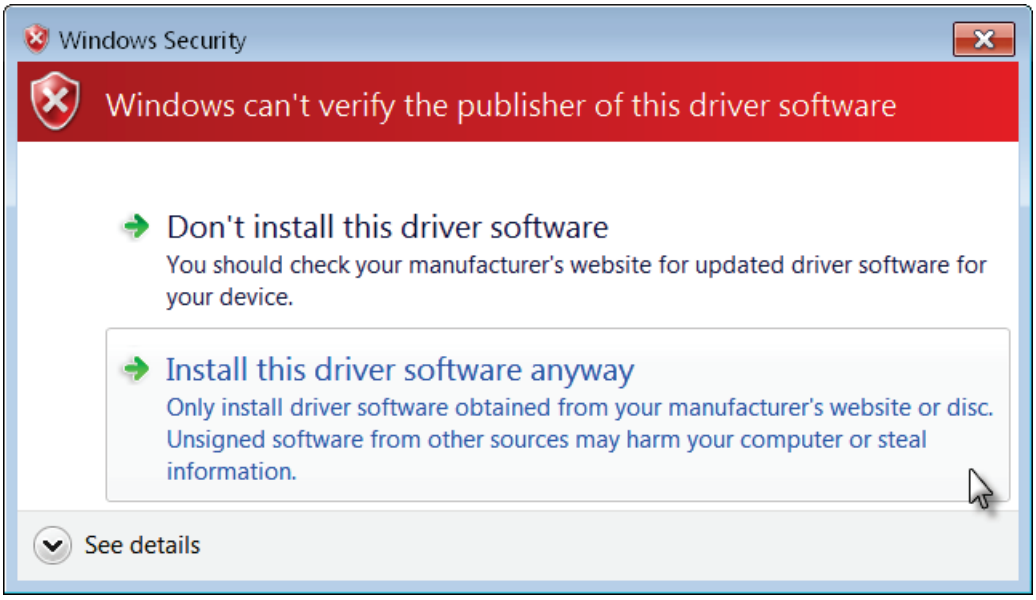

Windows Security
Windows can't verify the publisher of this driver software
Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.
Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.
See details

Figure 2.5 – Installing the driver for the Autolab (3/4)

Select the *Install this driver software anyway* option to proceed with the installation. At the end of the installation, a message will be displayed indicating that the driver has been successfully installed (see Figure 2.6).

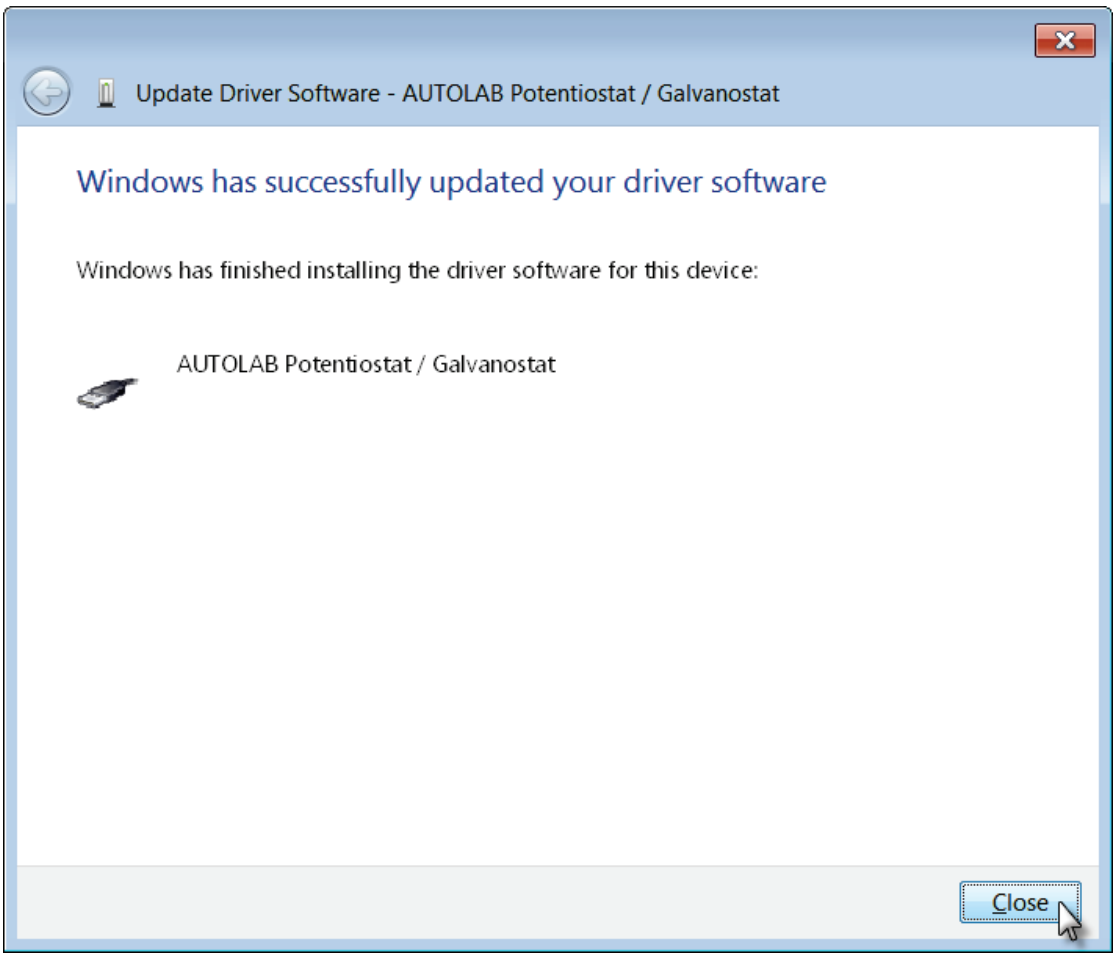

Update Driver Software - AUTOLAB Potentiostat / Galvanostat
Windows has successfully updated your driver software
Windows has finished installing the driver software for this device:
AUTOLAB Potentiostat / Galvanostat
Close

Figure 2.6 – Installing the driver for the Autolab (4/4)

2.1.2 – Windows XP

After Autolab SDK has been successfully installed, connect the Autolab instrument to the computer using an available USB port. Switch on the instrument. If no Autolab USB drivers are installed, the Found New Hardware window should appear (see Figure 2.7).

Note

If your computer is connected to the internet, you will be invited to go online to search for updated drivers for the new hardware. Click the *No, not this time* option and click the Next button to skip this part.

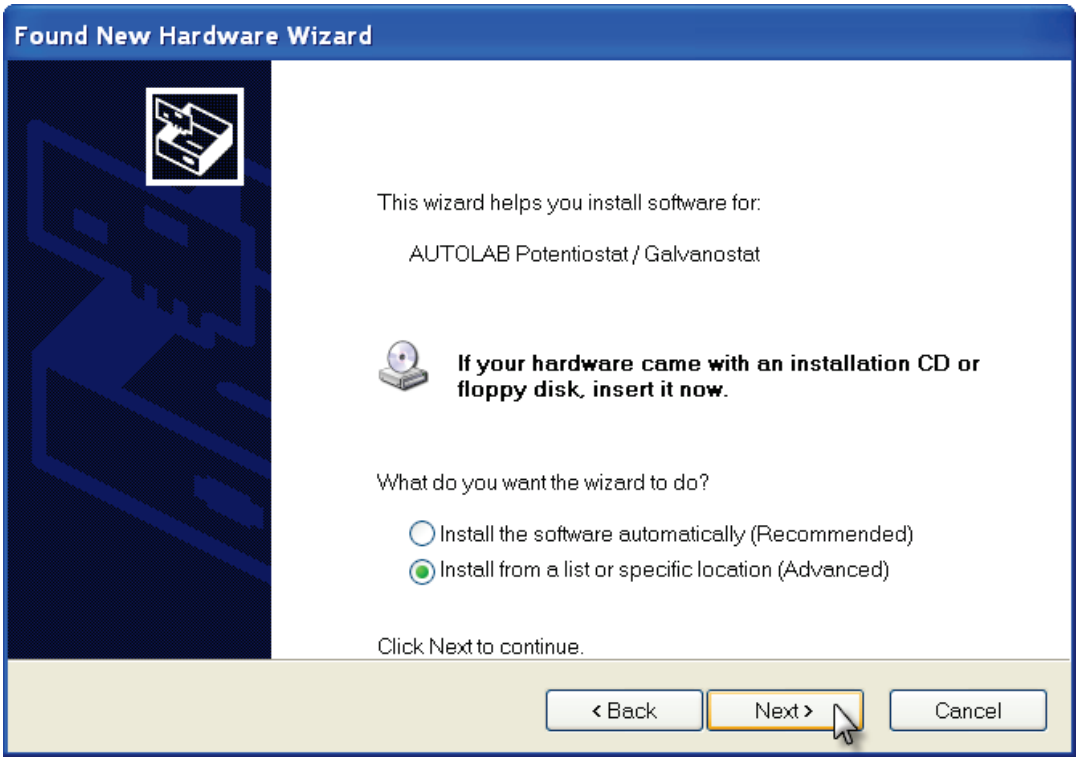

Found New Hardware Wizard
This wizard helps you install software for:
AUTOLAB Potentiostat / Galvanostat
If your hardware came with an installation CD or floppy disk, insert it now.
What do you want the wizard to do?
Install the software automatically (Recommended)
Install from a list or specific location (Advanced)
Click Next to continue.
< Back
Next >
Cancel

Figure 2.7 – The found new hardware window

Select the *Install from a list or specific location (Advanced)* option and click Next (see Figure 2.7). In the next screen, choose the *Search for the best driver in these locations* option and check the *Include this location in the search* (see Figure 2.8).

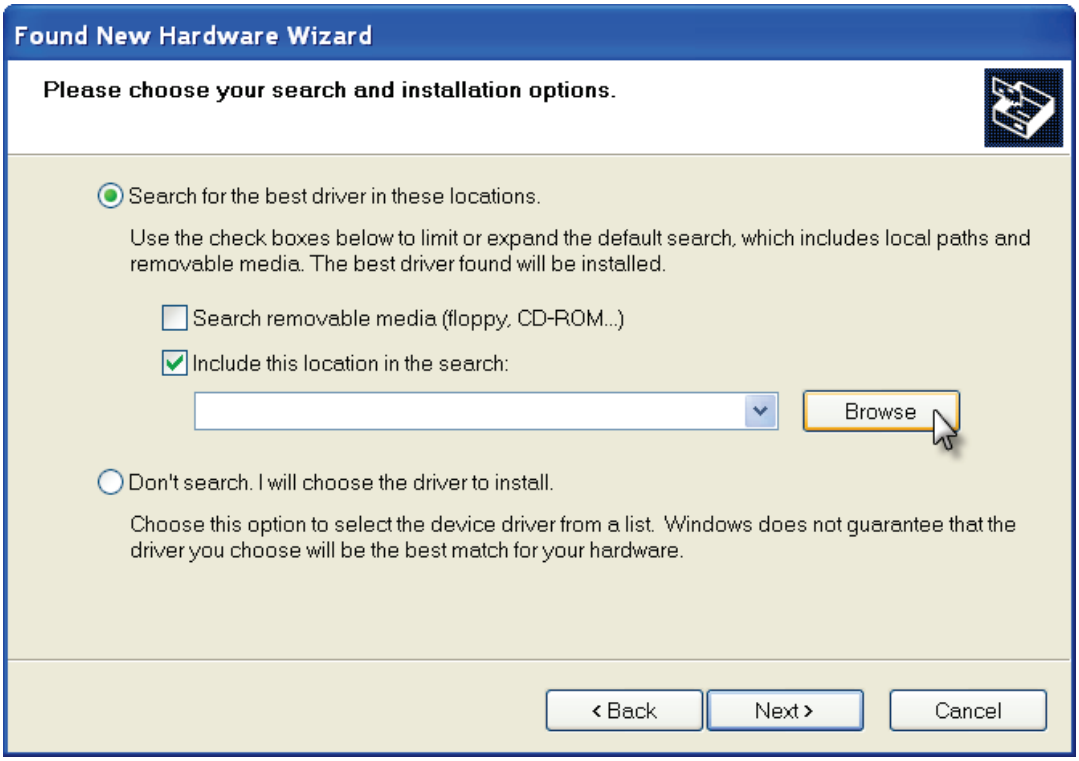

Found New Hardware Wizard
Please choose your search and installation options.
Search for the best driver in these locations.
Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.
Search removable media (floppy, CD-ROM...)
Include this location in the search:
Browse
Don't search. I will choose the driver to install.
Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.
< Back
Next >
Cancel

Figure 2.8 – Specifying the location of the drivers – part 1/3

Click the Browse button to specify the location of the Autolab drivers. In the Browse for Folder window, navigate to the *C:\Program Files\Common Files\Metrohm Autolab\Drivers\Usb* folder and click OK to continue (see Figure 2.9).

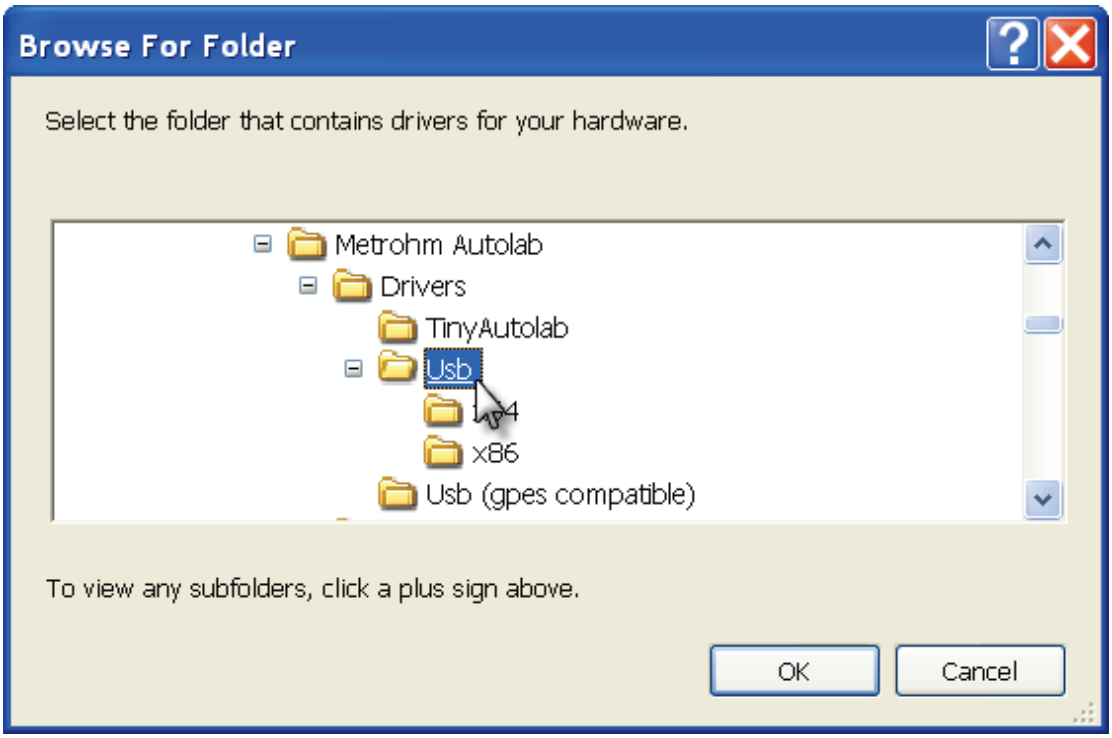


Figure 2.9 – Specifying the location of the drivers – part 2/3

Click Next to continue (see Figure 2.10).

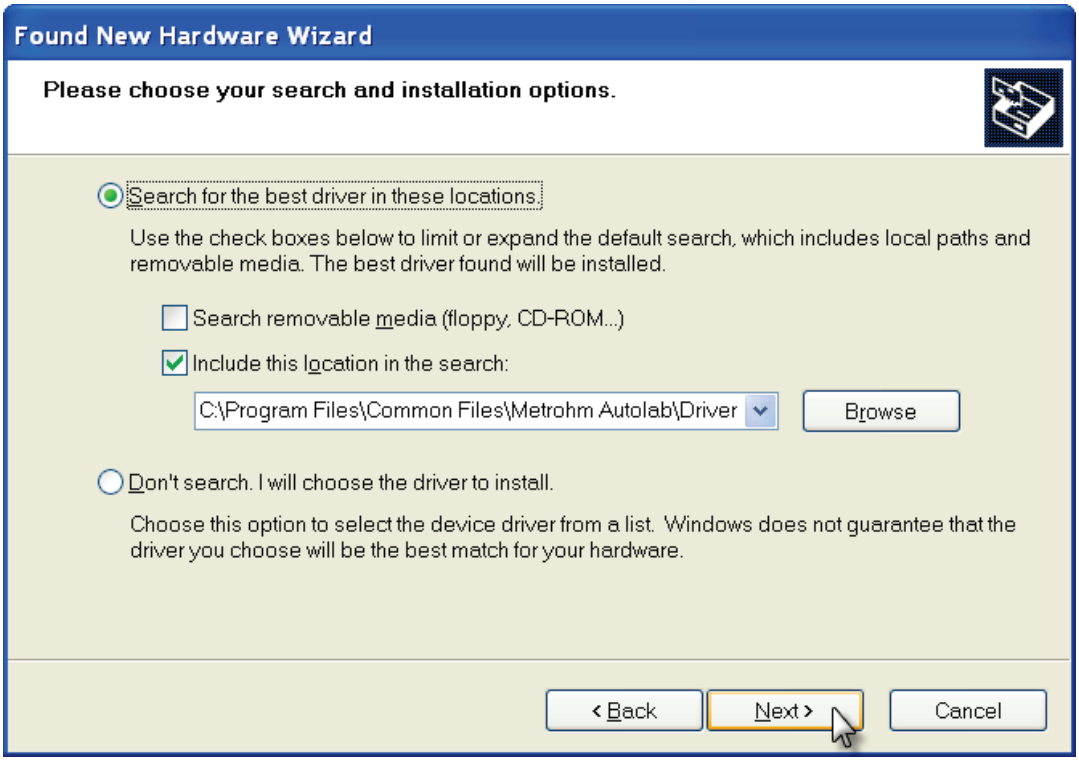

Found New Hardware Wizard
Please choose your search and installation options.
Search for the best driver in these locations.
Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.
Search removable media (floppy, CD-ROM...)
Include this location in the search:
C:\Program Files\Common Files\Metrohm Autolab\Driver
Browse
Don't search. I will choose the driver to install.
Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.
< Back
Next >
Cancel

Figure 2.10 – Specifying the location of the drivers – part 3/3

A warning message will appear, indicating that the selected driver has not passed Windows Logo tests (see Figure 2.11).

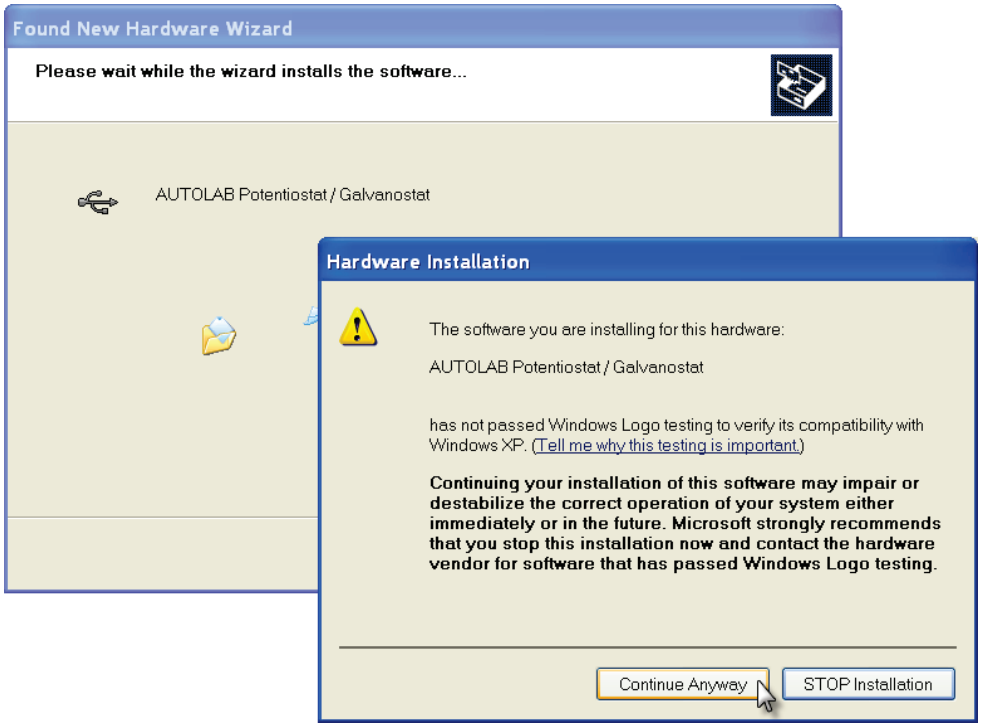

Found New Hardware Wizard
Please wait while the wizard installs the software...
AUTOLAB Potentiostat / Galvanostat
Hardware Installation
The software you are installing for this hardware:
AUTOLAB Potentiostat / Galvanostat
has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)
Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.
Continue Anyway
STOP Installation

Figure 2.11 – The compatibility warning

Click the Continue Anyway button to disregard this warning and complete the installation (see Figure 2.12).

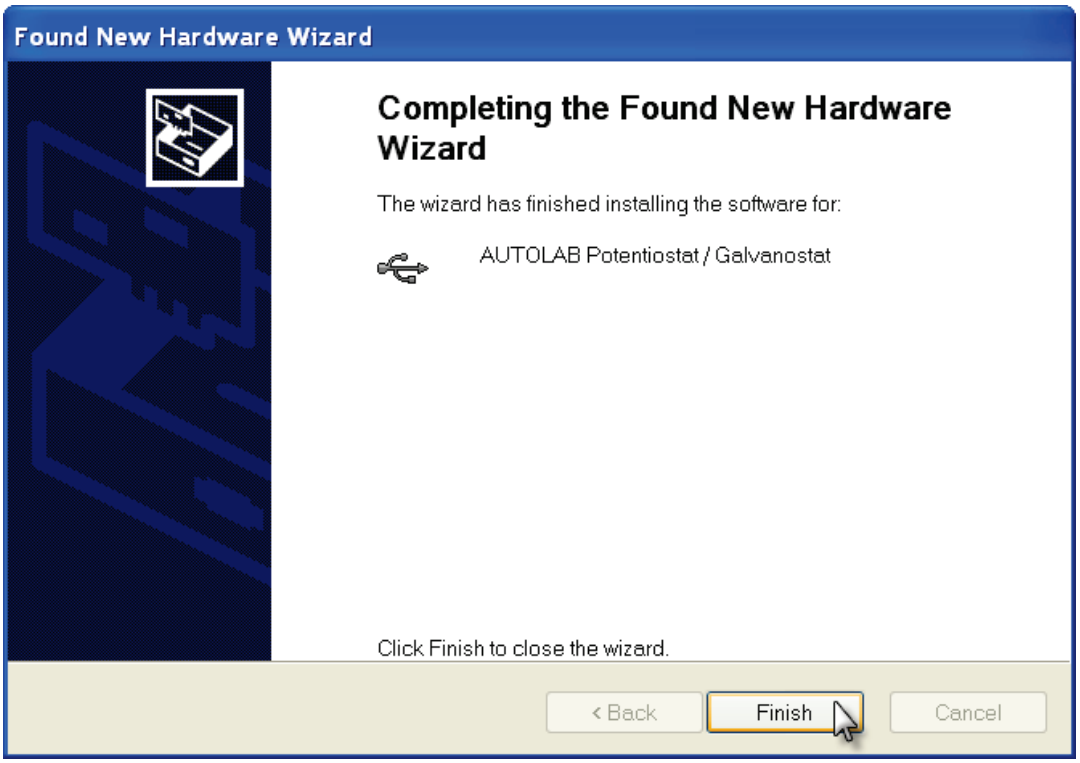

Found New Hardware Wizard
Completing the Found New Hardware Wizard
The wizard has finished installing the software for:
AUTOLAB Potentiostat / Galvanostat
Click Finish to close the wizard.
< Back
Finish
Cancel

Figure 2.12 – The drivers are correctly installed

3 – Autolab instrument

Before the Autolab instrument can be used with the Autolab SDK it must be connected to the computer. To use the Autolab instrument, the Autolab SDK requires an adk.x and a HardwareSetup.xml file.

Note

The **HardwareSetup.xml** file is required to configure the connected instrument properly. The most common hardware configurations are present in the `\Hardware Setup Files` folder. The **adk.x** file is the executable running on the embedded processor of the Autolab. It must be loaded and running before measurements can be performed with the instrument.

3.1 – Connect/Disconnect Autolab instrument

This section illustrates code to connect and disconnect an Autolab instrument:

```
//Create instance of your instrument.
Instrument MyAutolab = new Instrument()

//Enter the path to the HardwareSetup.xml file.
MyAutolab.HardwareSetupFile = "C:\...\HardwareSetup.xml"

//Enter the path to the adk.x file.
MyAutolab.AutolabConnection.EmbeddedExeFileToStart = "C:\...\adk.x"

//Connect Autolab.
MyAutolab.Connect()

//Insert here your code (DAC example).
MyAutolab.DAC.SetDac(3,3)

//When finished, disconnect Autolab.
MyAutolab.Disconnect()

End
```

Note

The code presented in the grey frame above (and in the rest of this document) is so-called **pseudocode**. The real code should be adapted to the syntax of the used programming language. This falls out of the scope of this manual.

The folder where the HardwareSetup.xml and adk.x file is located:

For Windows XP, Vista, WIndows 7 and Windows 8:

```
C:\Program Files\Metrohm Autolab\Autolab SDK 1.10\Hardware Setup
Files
```

Or, if the Nova setup must be used, the following paths are to be used:

For Windows XP:

```
C:\Documents and Settings\All Users\Application Data\Metrohm Autolab
```

For Windows Vista and Windows 7:

```
C:\ProgramData\Metrohm Autolab
```

4 – Procedure

With the Autolab SDK it is possible to load, measure and save procedures that are made with Nova. The Autolab SDK is suitable to:

- Measure a procedure.
- View measured data.
- Adjust procedure parameters.

A procedure is build with commands according the Nova procedure model.

4.1 – Procedure model

A procedure is a tree structured list of commands. A command can contain other command(s). The parameters of a command are stored in a list of command parameters as shown in Figure 4.1. The parameter can be input (for example measurement settings) or output (for example measured values).

The names of commands and command parameters consists of a user defined name and a key name.

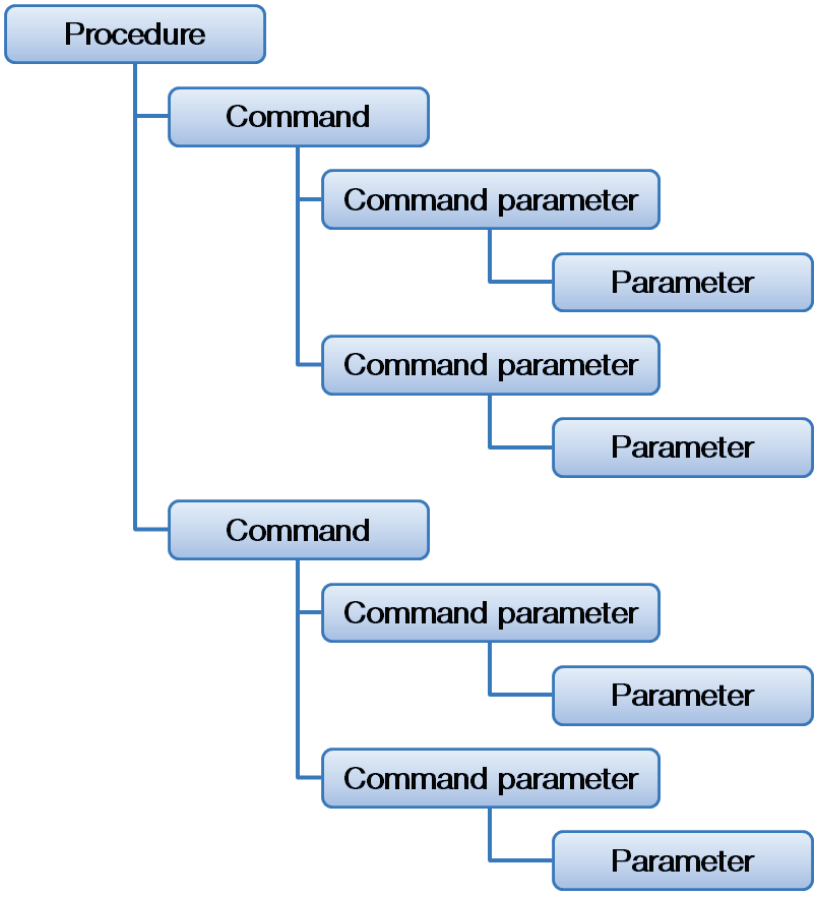


Figure 4.1 – Command model

Figure 4.2 shows a Cyclic voltammetry procedure in Nova:

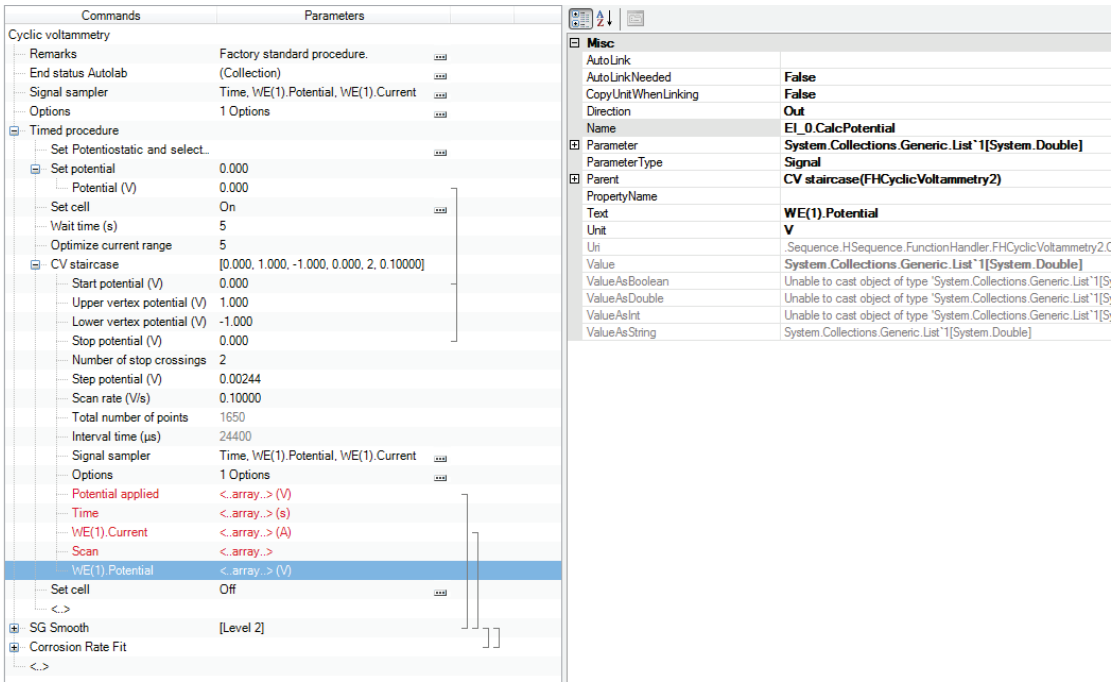


Figure 4.2 – Cyclic voltammetry procedure

Note

In order to distinguish between commands with the same name, change the description of this command. For instance with a procedure with two CV Staircase commands, rename the commands to *CV Staircase 1* and *CV Staircase 2*.

4.2 – Using procedures

This section illustrates code to load, measure and save a procedure. To load a procedure, type the path where the procedure is located in the LoadProcedure function.

```
//Create instance of your instrument.
Instrument MyAutolab = new Instrument()

//Enter the path to the HardwareSetup.xml file
//(see '2.1 Connect/Disconnect Autolab Instrument').
MyAutolab.HardwareSetupFile = "C:\...\HardwareSetup.xml"

//Enter the path to the adk.x
//(see '2.1 Connect/Disconnect Autolab Instrument').
MyAutolab.AutolabConnection.EmbeddedExeFileToStart = "C:\...\adk.x"

//Connect Autolab.
MyAutolab.Connect()

//Insert here your code.
//Load procedure.
Procedure myProcedure = MyAutolab.LoadProcedure("C:\Procedure.nox")

//Measure procedure.
myProcedure.Measure()

//Wait till the measurement is finished.
//When the measurement is finished, save the measured procedure.
myProcedure.SaveAs("C:\MeasuredProcedure.nox")

//When finished, disconnect Autolab.
MyAutolab.Disconnect()

End
```

The Autolab SDK is delivered with standard procedures, these are located in:

For Windows XP, Vista, Windows 7 and Windows 8:

```
C:\Program Files\Metrohm Autolab\Autolab SDK 1.10\Standard Procedures
```

This section illustrates code to view the measured values of a procedure. The Autolab SDK uses the command model of the procedure to get the data.

```
//Create instance of your instrument.
Instrument MyAutolab = new Instrument()

//Enter the path to the HardwareSetup.xml file
//(see '2.1 Connect/Disconnect Autolab Instrument').
MyAutolab.HardwareSetupFile = "C:\...\HardwareSetup.xml"

//Enter the path to the adk.x
//(see '2.1 Connect/Disconnect Autolab Instrument').
MyAutolab.AutolabConnection.EmbeddedExeFileToStart = "C:\...\adk.x"

//Connect Autolab.
MyAutolab.Connect()

//Insert here your code.
//Load procedure.
Procedure myProcedure = MyAutolab.LoadProcedure("C:\Procedure.nox")

//Select command due to its command name or key name
//(see attachment '5.1 Command names').
Command myCommand = myProcedure.Commands["FHCyclicVoltammetry2"]

//Select output command parameter due to its command parameter name
//or key name, in this example the Potential
//(see '5.3 Output command parameter names').
//And cast it to the associated type
//(see '5.4 Command parameter types').
CommandParameterDoubleList potential =
(CommandParameterDoubleList)myCommand.Signals[Potential]

//Get the value of the parameter.
Double[] MeasuredPotential = potential.Value

//When finished, disconnect Autolab.
MyAutolab.Disconnect()

End
```

This section illustrates code to adjust procedure input parameters. The Autolab SDK uses the command model of the procedure to set these parameters.

```
//Create instance of your instrument.
Instrument MyAutolab = new Instrument()

//Enter the path to the HardwareSetup.xml file
//(see '2.1 Connect/Disconnect Autolab Instrument').
MyAutolab.HardwareSetupFile = "C:\...\HardwareSetup.xml"

//Enter the path to the adk.x
//(see '2.1 Connect/Disconnect Autolab Instrument').
MyAutolab.AutolabConnection.EmbeddedExeFileToStart = "C:\...\adk.x"

//Connect Autolab.
MyAutolab.Connect()

//Insert here your code.
//Load procedure.
Procedure myProcedure = MyAutolab.LoadProcedure("C:\Procedure.nox")

//Select command due to its command name or key name
//(see '5.1 Command names" for example names').
Command myCommand = myProcedure.Commands["FHCyclicVoltammetry2"]

//Select input command parameter due to its command parameter name
//or key name, in this example the Upper vertex
//(see '5.2 Input command parameter names').
//And cast it to the associated type
//(see '5.3 Command parameter types').
CommandParameterDouble UpperVertex =
(CommandParameterDouble)myCommand.Signals[Upper vertex]

//Set the value of the parameter.
UpperVertex.Value = 2

//When finished, disconnect Autolab.
MyAutolab.Disconnect()

End
```

This section illustrates code to monitor the potential during the measurement. It is also possible to monitor the current and time. These values can not be monitored for a FRA measurement.

```
//Create instance of your instrument.
Instrument MyAutolab = new Instrument()

//Enter the path to the HardwareSetup.xml file
//(see '2.1 Connect/Disconnect Autolab Instrument').
MyAutolab.HardwareSetupFile = "C:\...\HardwareSetup.xml"

//Enter the path to the adk.x
//(see '2.1 Connect/Disconnect Autolab Instrument').
MyAutolab.AutolabConnection.EmbeddedExeFileToStart = "C:\...\adk.x"

//Connect Autolab.
MyAutolab.Connect()

//Insert here your code.
//Load procedure.
Procedure myProcedure = MyAutolab.LoadProcedure("C:\Procedure.nox")

//Measure procedure.
myProcedure.Measure()

//Get the potential, current or time from the sampler
//(WE(1).Potential, WE(1).Current or Time).
Signal potential = MyAutolab.EI.Sampler.GetSignal("WE(1).Potential")

//Get the value of the potential.
Double Potential = potential.Value

//When finished, disconnect Autolab.
MyAutolab.Disconnect()

End
```

5 – LabVIEW

This chapter explains the use of the Autolab SDK in LabVIEW.

The set up in LabVIEW consist of three parts:

- Connect Autolab instrument.
- Adjust procedure parameters, measure procedure, view measured data, and/or control Autolab modules.
- Disconnect Autolab instrument.

5.1 – Install LabVIEW examples

To use the Autolab SDK in LabVIEW install the *Autolab SDK* according to Chapter 2.

The Autolab SDK package is delivered with LabVIEW examples. The dll's and the project (Autolab.lvproj) with the SubVI's and Examples are located in:

```
C:\Program Files\Metrohm Autolab\Autolab SDK 1.10
```

The LabVIEW example project is build up as follows:

- **SubVI:** General used SubVI's for i.e. Connecting and Releasing Autolab instruments.
- **Advanced SubVI:** SubVI's used in the Advanced examples. Can be used as building blocks for customized LabVIEW applications.
- **Basic SubVI:** SubVI's used in the Basic examples. Can be used as building blocks for customized LabVIEW applications.
- **Advanced Examples:** Example application using the Advanced SubVI´s. The Hardware setup as well as the embedded executable and measurement procedures must be specified manually for each example.
- **Basic Examples:** Example application using the Basic SubVI´s. When the Hardware setup is chosen properly, these examples are ready to run. Standard the Hardware setup is configured for a PGSTAT302N with FRA2.

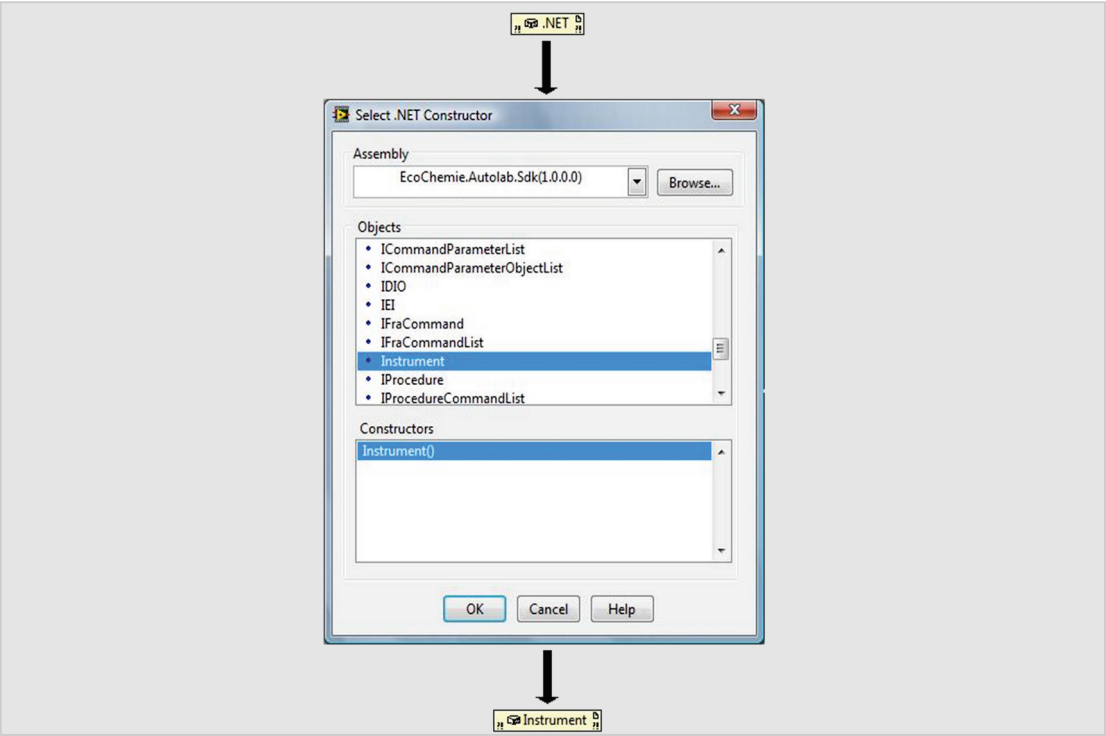
5.2 – Using Autolab SDK in LabVIEW

To use the Autolab SDK in LabVIEW install the Autolab SDK according to Chapter 2.

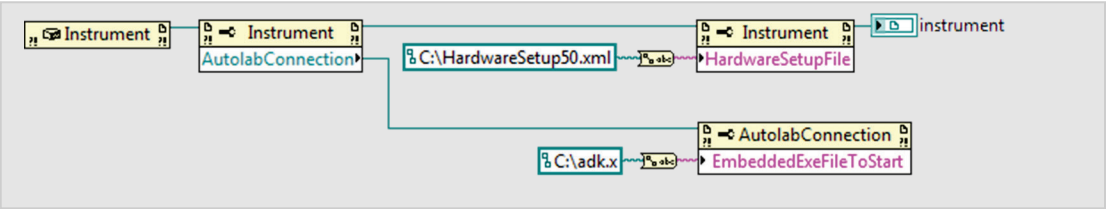
The explanation illustrates a part of a Cyclic voltammetry (CV) measurement. In this explanation the Autolab SDK is used to:

- Connect Autolab instrument.
- Adjust the procedure parameter *Upper vertex voltage*, measure the procedure, monitor potential during measurement and view the measured current.
- Disconnect Autolab instrument.

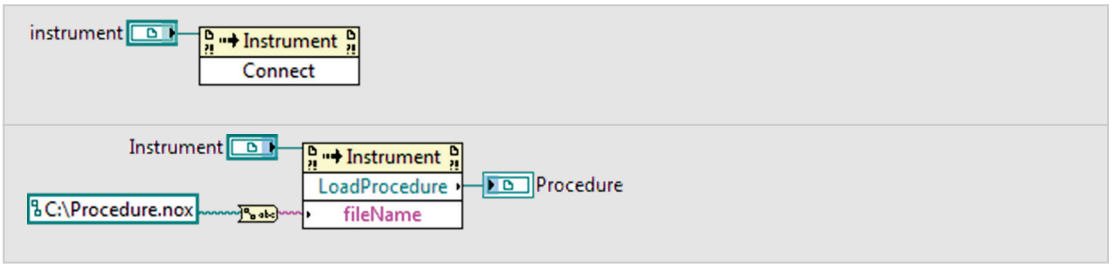
To integrate the Autolab SDK, place a .NET constructor on the block diagram and select Instrument from the EcoChemie.Autolab.Sdk.dll assembly (`C:\Program Files\Metrohm Autolab\Autolab SDK 1.10`)



Enter the path to the adk.x and HardwareSetup.xml file (see Section 2.1):



Connect Autolab instrument:



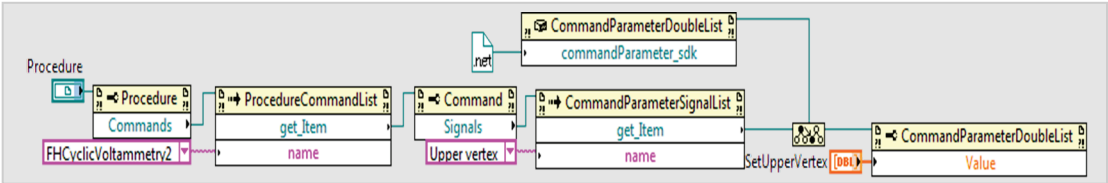
Load procedure:

Adjust the Upper vertex voltage:

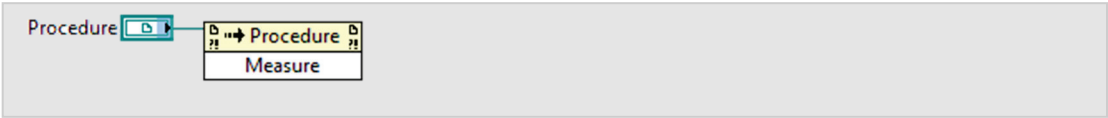
- Select CV staircase command (FHCyclicVoltammetry2);
- Select the upper vertex command parameter (Upper vertex);
- Use "to more specific type (CommandParameterDoubleList)";
- Set the value.

Note

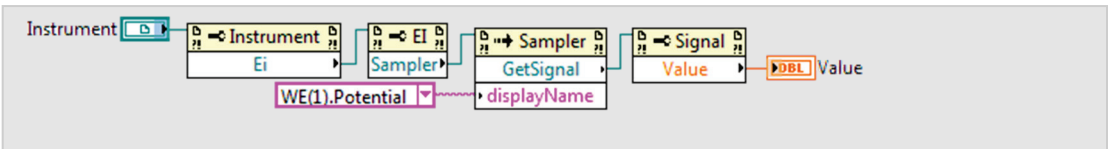
See Sections 6.1-6.4 for more information on the name of the commands and command parameters.



Measure procedure:



To view the potential during the measurement, get the potential (WE(1).Potential) from the sampler:



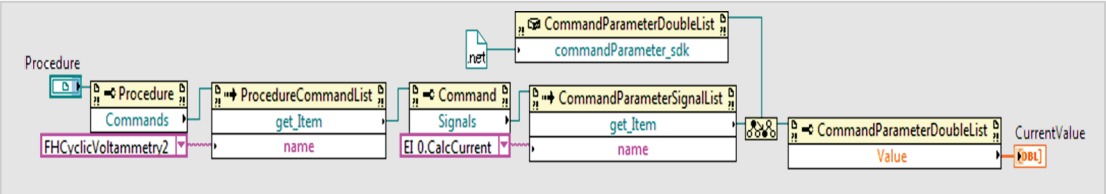
To view the measured current:

- Select CV staircase command (FHCyclicVoltammetry2);

- Select command parameter (EI_0.CalcCurrent);
- Use "to more specific type (CommandParameterDoubleList)";
- Get the value.

Note

See Sections 6.1-6.4 for more information on the name of the commands and command parameters.



When finished, disconnect Autolab:

6 – Appendix

6.1 – Command names

To select a command in the Autolab SDK it is possible to use the command name, which can be changed in Nova, or to use the key name, this is a fixed general name. Switch on "Advanced" in Nova to see the names (see Figure 6.1).

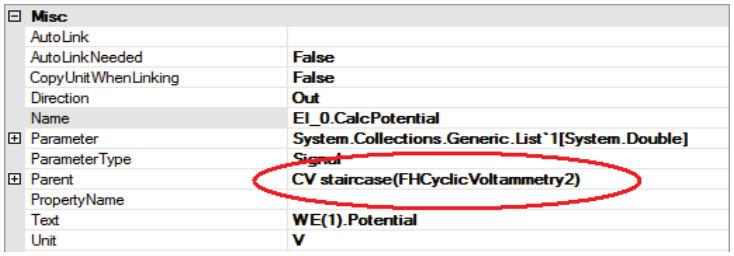


Figure 6.1 – The name and key of a command is shown in the advanced view of NOVA

Example command names and their key names:

Command name	Command key
CV staircase	FHCyclicVoltammetry2
LSV staircase	FHLinearSweep
CV linear scan	CVLinearScanAdc164
CV linear scan high speed	CVLinearScanAdcHs
CV staircase galvanostatic	FHCyclicVoltammetryGalvanostatic
LSV staircase galvanostatic	FHLinearSweepGalvanostatic
Record signals (> 1 ms)	FHLevel
Chrono methods	RecordLevelsContainer
Chrono methods high speed	HighSpeedLevelsContainer
Chrono methods high speed galvanostatic	HighSpeedLevelsContainer
FRA frequency scan potentiostatic	FIAScan
FRA frequency scan galvanostatic	FIAScan
FRA frequency scan external	FIAScan
Corrosion rate, fit	CorrosionRateFitCommand
Corrosion rate, Tafel slope	CorrosionRateTafelSlopeCommand
SG smooth	MathSmoothSavitzkyGolay
Peak search	PeakSearchCommand
Nested procedure	ExecCommandSequence
Calculate signal	MathParser
Windower	SignalWindowerCommand
Scan selector	SignalWindowerCommand

6.2 – Input command parameter names

To select an input command parameter in the Autolab SDK it is possible to use the command name, or to use the key name.

Example Cyclic voltammetry input command parameter names and key names:

Parameter name	Parameter key
Start value (V)	Start value
Upper vertex potential (V)	Upper vertex
Lower vertex potential (V)	Lower vertex
Step potential (V)	Step
Interval time (s)	Interval time
Number of stop crossings	NrOfStopCrossings
Stop potential (V)	Stop value
Scan rate (V/s)	Scanrate

6.3 – Output command parameter names

To select an output command parameter in the Autolab SDK it is possible to use the command name, or to use the key name.

Example Cyclic Voltammetry output command parameter names and key names:

Parameter name	Parameter key
Potential	EI_0.CalcPotential
Current	EI_0.CalcCurrent
Time	CalcTime
Potential applied	SetpointApplied

6.4 – Command parameter types

To cast the command parameter to the associated type the following types are available in the Autolab SDK:

- CommandParameterBool (returns bool type)
- CommandParameterDouble (returns double type)
- CommandParameterInt (returns int type)
- CommandParameterOnOff (returns double array type)
- CommandParameterDoubleList (returns enum type)

6.5 – General remarks using individual modules

General remarks on the use of Autolab SDK application to control individual modules:

- In order to avoid spikes during the measurement it is advised not to use the AD converter of the Autolab instrument during measurement with an external data acquisition board.
- Connection of an Autolab instrument to an external Data acquisition system can introduce ground loops in the measurement system. This can introduce higher noise levels than expected. The shields of the BNC connectors are connected together inside the Autolab instruments. If possible, avoid analog ground connections between the AD and DA converters on the data acquisition board.
- Applying signals to the Potentiostat/Galvanostat system of the Autolab is restricted due to bandwidth limitations. The overall bandwidth is defined by the control loop. For the Autolab instruments this is:
    - In ultra High Speed mode:
        - 500 kHz for PGSTAT128N, 12, 100N, 100 and µAutolabIII
          1 MHz for PGSTAT30, 302, 302N, PGSTAT101/204 and M101
    - In High Speed mode:
        - 100 kHz
    - In High Stability mode:
        - 10kHz

Furthermore the bandwidth is limited by the measurement part. The current follower is limited per current range as follows:

Current range	PGSTAT30/302/ 302N/128N/302F	PGSTAT12/100/ 100N
1 A	5 MHz	
100 mA	5 MHz	5 MHz
10 mA	5 MHz	5 MHz
1m A	5 MHz	5 MHz
100 µA	1 MHz	1 MHz
10 µA	100 kHz	100 kHz
1 µA	10kHz	10 kHz
100 nA	1 kHz	1 kHz

Current range	µAutolab	PGSTAT204	PGSTAT101/M101
100 mA		5 MHz	
10 mA	5 MHz	5 MHz	3 MHz
1m A	5 MHz	5 MHz	3 MHz
100 µA	500 kHz	5 MHz	1 MHz
10 µA	50 kHz	10 kHz	75 kHz
1 µA	5 kHz	500 Hz	20 kHz
100 nA	400 Hz	500 Hz	4 kHz

Example1: Apply square wave of 500Hz, µAutolab_III:

- Control loop: High Stability
- Current range 10mA-1µA (not lower). The 1µA current range will already start to give lower currents then expected.

Example2: Apply Sine wave of 75kHz, µAutolab_III:

- Control loop: High Speed
- Current range 10mA-100µA (not lower).

- The Ein input (if available) is a 1:1 input. So the Voltage set on this input will be the voltage on the cell. Take care that a voltage step on the Potentiostat will lead to immediate electrochemical response. The best sequence to switch on the cell is:

  1. apply a potential (or current)
  2. switch the cell on
  3. start the pre-treatment and measurement.

- When the Galvanostat is used, the applied current is equal to the Applied voltage multiplied by the selected current range. So 1 V on the 1 mA current range means an applied current of 1mA.
- The Eout is a 1:-1 output. So .5 V on Eout means -.5 V over the Electrochemical cell.
- The Iout is a 1:1*c.r. output. That means that 1 V output on the 1 mA current range is equal to 1 mA current trough the cell.
- The Manual Cell status in the Manual Control test.vi example is not available for the µAutolab type II and III.
- The setpoint of the PGSTAT must be set to zero volt to avoid unexpected offsets. Setting the DAC of the Autolab will lead to extra offset. The voltage of the setpoint and the Ein will be added up. So a setpoint of .5 V with an Ein of .3 V will lead to .8 V setpoint on the electrochemical cell.

Acknowledgements

Metrohm Autolab gratefully acknowledges the following following people for their scientific contribution in LabVIEW software development:

Prof. Dr. Doru Ursutiu - udoru@unitbv.ro
Sef. Lucr. DrD Petru Cotfas - pcotfas@unitbv.ro
Center for Valorization and Transfer of Competence CVTC
University Transylvania of Brasov - Romania
B- dul Eroilor 29, RO- 500036, Brasov, Romania
Tel/ Fax: +40 (0)268-415213
Mobile: +40 (0)744-756640

Autolab SDK User manual

03/2013

Kanaalweg 29/G
3526 KM Utrecht
The Netherlands

Metrohm
Autolab B.V.